AUTOMATIC SCRIPT GENERATION USING DEEP LEARNING

Milestone-1 Presentation

Komal Atul Sorte
Student
ks7229@rit.edu

Prof. Hans Peter Bischof
Faculty Advisor
hpb@cs.rit.edu

Prof. Matthew Fluet
Course Advisor
mff@cs.rit.edu
Target

Build and enable an AI agent such that it learns a piece of script and automatically generates text based on appropriate context.

Further Accuracy Improvement

By considering,

- Contextual meaning of words
- Synonyms (e.g. check or validate)
- Word embeddings
Deliverables

WEEK 4

Backend
Analytical Data

Research
Finalize data source
*F.R.I.E.N.D.S TV series*

Design solution framework
Web crawl
Data Pre-processing

**CHALLENGES**

Changed data source from *The Office* to *F.R.I.E.N.D.S*

Did not remove frequent words.
Tense
Solution Architecture
Data Pre-processing
1. Import the data and relevant libraries for data manipulation
2. Perform high level data inspection followed by detailed exploratory data analysis
3. Conduct data cleaning and treatment to get data in consumable format
4. Perform text analysis

Web Scraping
1. Search and identify apt data sources and elements
2. Understand how the data is organized and fetched on/from the source
3. Develop a crawler to extract data elements from source
4. Import data into the storage system in a form which could be further processed

Building AI Agent
- Final Data
- Divide data into train, validation and test
- 60%
- 20%
- 20%
- Training Data Scripts
- Validation Data Scripts
- Testing Data Scripts

Prediction
- Input Line
- Model is ready!
- Must predict and write a script
- Output Text/Dialogue

Note: LSTM and Fine Tuning are part of the machine learning process.
Web Crawl
Gather Data

Data Source

**F.R.I.E.N.D.S.**

**Technique - Web Scraping**
- Systematically find & download web pages.
- Extract information from them.
- Data stored in .txt files

**Technology**
- Python, BeautifulSoup library
Scripts
Pre-processing
Clean Scripts

Remove unnecessary details: E.g. Written By:, Title:, Additional Notes, etc.
Remove blank lines

Create Batches

Reason: To cater the problem of resources.
4 Batches

Reduce Vocabulary Size

Lowercase: E.g. SKY, Sky, sky all mean the same
Remove Punctuations: E.g. friday!!, friday?, friday mean the same

Word/ Character Level Embeddings

Generate Tokens: A list of words/ characters
Create Sequences from tokens: Slides a window over given seed length.

Store Data

Save sequences into database

Model will be trained to understand,
Probability of 76th word/char 1
Given the sequence of last 75 words/chars in order.
Word Embeddings

- For semantically correct output

“21892 Unique Words”
Character Embeddings

- For syntactically & semantically correct output
- Any word's vector can be determined even if out of vocabulary.

"45 Unique Characters"
Next Steps
Milestones

Milestone 1:
- Backend analytical data is created

Milestone 2:
- Final model is ready

Milestone 3:
- Final API is ready, and model is tested

Final:
- Final Presentation & Poster
Milestone 2

**Build Model**

- Encodings - Mapping of all elements from our vocab to an integer value
- Train model over as much data as possible.
- Model must understand:
  
  **Probability of 76th word/char given the sequence of last 75 words/chars in order.**

**Possible Challenges**

- Over fitting of the model - Hyper Parameters
- Which proves better - word/character level embedding?
- Determining how model works after combining both.
- Represent synonyms as one entity. - *Embedding layer*
References


Movie Sunspring - https://www.youtube.com/watch?v=LY7x2Ihqjmc

Dataset - https://fangj.github.io/friends/
Thank You